**SPECTACOLE – GREEDY -** MILOSESCU

#include <iostream> #include <fstream> using namespace std;

ifstream fin ("date.in"); ofstream fout ("date.out");

struct activitate { int x, y, k; // x = ora de inceput, y = ora de sfarsit, k=nrul activitatii, un soi de cod };

activitate a[20]; int n, i, j , m, s[20];

void **citeste** () { fin>>n;

for (i=1; i<=n; i++) { fin>>a[i].x>>a[i].y; a[i].k=i; } fin.close(); }

void **sort**() { for (i=1; i<=n-1; i++) for (j=i+1; j<=n; j++) if (a[i].y>a[j].y) swap(a[i], a[j]); }

void **greedy** () { s[1]=1; j=1;

for (i=2; i<=n; i++)

if( a[i].x >= a[s[j]].y ) { j++; s[j]=i; }

m=j; }

void **afiseaza**() { fout<<"Planificarea activitatilor: "<<endl;

for (i=1; i<=m; i++)

fout<<"Activitatea "<<a[s[i]].k<<" incepe la ora "<<a[s[i]].x<<" si se term la ora "<<a[s[i]].y<<endl;

fout.close(); }

int main() { citeste(); sort(); greedy(); afiseaza(); return 0; }

|  |  |
| --- | --- |
| DATE.IN | DATE.OUT |
| 8  9 11  12 13  8 10  10 12  16 18  14 16  20 22  19 21 | Planificarea activitatilor:  Activitatea 3 incepe la ora 8 si se term la ora 10  Activitatea 4 incepe la ora 10 si se term la ora 12  Activitatea 2 incepe la ora 12 si se term la ora 13  Activitatea 6 incepe la ora 14 si se term la ora 16  Activitatea 5 incepe la ora 16 si se term la ora 18  Activitatea 8 incepe la ora 19 si se term la ora 21  Sol este 3 4 2 6 5 8 |

SORT – crescator dupa timpul de terminare a activitailor ( campul y)

GREEDY – depun in vectorul solutie primul element din a (activitatea cu ora de terminare cea mai mica)

Pentru urm. n-1 activit executa: daca ora la care incepe activitatea curenta >= ora a care se termina activitatea ce a fost adaugata ultima la solutie (in vectorul s) atunci activitatea este adaugata la solutie.

**RTUCSAC – GREEDY -** MILOSESCU

#include <iostream> #include <fstream> using namespace std; ifstream fin ("date.in"); ofstream fout ("date.out");

struct obiect { int k; float g, c, e; // g - greutate c - profit, k - nrul obiectului e – eficienta };

obiect a[20], aux; int n,i, j, m, s[20]; float G, Gr, x[20];

void citeste () { fin>>n>>G; for (i=1; i<=n; i++) { fin>>a[i].g>>a[i].c; a[i].k=i; a[i].e=a[i].c/a[i].g; }

fin.close(); }

void sort() { for (i=1; i<=n-1; i++) for (j=i+1; j<=n; j++) if (a[i].e<a[j].e) swap(a[i], a[j]); }

void greedy ()

{

j=0; Gr=G;

for (i=1; i<=n && Gr!=0; i++)

if(Gr>a[i].g) { j++; s[j]=i; x[j]=1; Gr=Gr-a[i].g; }

else {j++; s[j]=i; x[j]=Gr/a[i].g; Gr=0; }

m=j; }

void afiseaza() { for (i=1; i<=m; i++)

fout<<"Obiectul "<<a[s[i]].k<<" in cantitatea "<<x[i]\*a[s[i]].g<<endl; fout.close(); }

int main() { citeste(); sort(); greedy(); afiseaza(); return 0; }